



# Smart Hospital Management System

Rakesh Yadav<sup>1</sup>, Arvind Yadav<sup>2</sup>, Bhanu Chandra<sup>3</sup>

Student, B.Tech CSE, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh, India<sup>1-3</sup>

**Abstract:** This paper details the design, development, and features of a web-based Hospital Management System (HMS) aimed at streamlining patient registration, appointment scheduling, and administrative oversight. The system utilizes a three-module architecture (Patient, Doctor, Admin) providing distinct interfaces and functionalities for each user type. Built using HTML, CSS, JavaScript, and Bootstrap for the front-end, and PHP with a MySQL RDBMS for the back-end logic and data storage, the system offers features including user registration/login, appointment booking and cancellation, appointment history viewing, doctor management (addition/removal by admin), patient/doctor searching, and feedback viewing. The system offers an easy interface and demonstrates a real-world use of standard web technologies for managing basic healthcare needs. We have implemented this project in project-based learning.

**Keywords:** Hospital Management System, HMS, Web Application, PHP, MySQL, Bootstrap, Appointment Scheduling, Patient Management, Healthcare IT, Modular Design, Database Management

## I. INTRODUCTION

Efficient management systems are fundamental to the effective functioning of contemporary healthcare facilities, helping to overcome the significant drawbacks inherent in traditional paper-based approaches, which often lead to operational inefficiencies, a higher likelihood of errors, and considerable difficulty in retrieving patient information promptly. This context underscores the clear need for accessible, cost-effective, and intuitive digital systems designed to streamline the essential interactions between patients, medical practitioners, and administrative personnel, with a specific focus on simplifying patient registration and the complexities of appointment scheduling. To address this requirement, this paper introduces a newly developed web-based Hospital Management System (HMS), architected with three distinct modules – Patient, Doctor, and Admin – each providing role-specific functionalities. The core objectives of this project were to design and implement a secure web application using PHP, MySQL, HTML/CSS/JS, and Bootstrap to enable online patient registration and login, allow patients to conveniently book, view, and cancel their appointments, permit doctors to access and manage their appointment schedules, and empower administrators with oversight functions including user management and feedback review. While the system's scope concentrates on these vital registration, appointment, and basic user management tasks (excluding features like EMR, billing, or inventory), it provides a functional foundation for improved hospital workflow. The subsequent sections of this paper will delve into related works, detail the system's design and implementation, evaluate its features, discuss its implications and limitations, and finally, offer conclusions and suggestions for future enhancements.

### Ease of Use

The Hospital Management System was designed with user-friendliness as a key priority. Its web-based interface, built using Bootstrap, offers a clean and familiar layout that is easy to navigate on different devices.

By separating functions into distinct modules for patients, doctors, and administrators, the system presents only relevant options to each user group, simplifying workflows.

Core tasks like registration, booking appointments, and viewing information are straightforward, making the system intuitive and accessible for its intended users with minimal difficulty.

### Objectives:

- To design and develop a secure web application for hospital management.
- To implement distinct modules for patients, doctors, and administrators with role-specific functionalities.
- To facilitate online patient registration and login.
- Allow patients to schedule, check, and cancel their appointments through an online platform.
- To allow doctors to view and manage their appointment schedules.
- To provide administrators with oversight capabilities, including user management (patients, doctors), appointment viewing, and feedback management.
- To utilize PHP, MySQL, HTML/CSS/JS, and Bootstrap for development.

### Scope:



The system focuses on patient registration, appointment management (booking, viewing, cancellation), basic doctor management, and feedback viewing. It does not include features like electronic medical records (EMR), billing, pharmacy management, or inventory control.

## II. LITERATURE SURVEY

HMS have evolved from manual to web-based systems to improve efficiency and accessibility in healthcare. Web-based HMS offer benefits like scalability and ease of access but face challenges security and data privacy (Smith et al., 2020).

PHP and MySQL are commonly used for server-side development and database management in such systems due to their flexibility and community support (Jones, 2018). Front-end technologies like HTML5, CSS3, and JavaScript enhance user interfaces, with frameworks like Bootstrap enabling responsive design (Brown, 2022). JavaScript enables websites to update content without needing to reload, enhancing the user's experience (Williams, 2019). PDF generation using libraries like TCPDF is useful for reports and documentation (Davis, 2021).

Security and privacy are critical concerns, requiring adherence to regulations like HIPAA and GDPR (European Union, 2016; Health Insurance Portability and Accountability Act, 1996). Best practices in web development are essential to mitigate threats like SQL injection and XSS (Miller, 2023). User-centered UI/UX design is crucial for system adoption and efficiency across different user roles (Green, 2024).

Existing research explores various modules within HMS, including patient registration, appointment booking, and administrative functions (Lee, 2017). However, there is ongoing work in areas like seamless integration of these modules and enhanced security protocols for web-based platforms. This research aims to contribute by developing a comprehensive, user-friendly HMS utilizing PHP, MySQL, Bootstrap, and JavaScript, while addressing key security and functional requirements for patients, doctors, and administrators.

## III. SYSTEM DESIGN AND ARCHITECTURE

**3.1 Class diagram:** A class diagram for a hospital management system is a visual representation of the system's structure, showing the different categories of data (classes) such as patients, doctors, appointments, and hospital departments. It outlines the characteristics (attributes) and functions (methods) of each class and illustrates how these classes are connected and interact with each other to manage hospital operations and patient care.

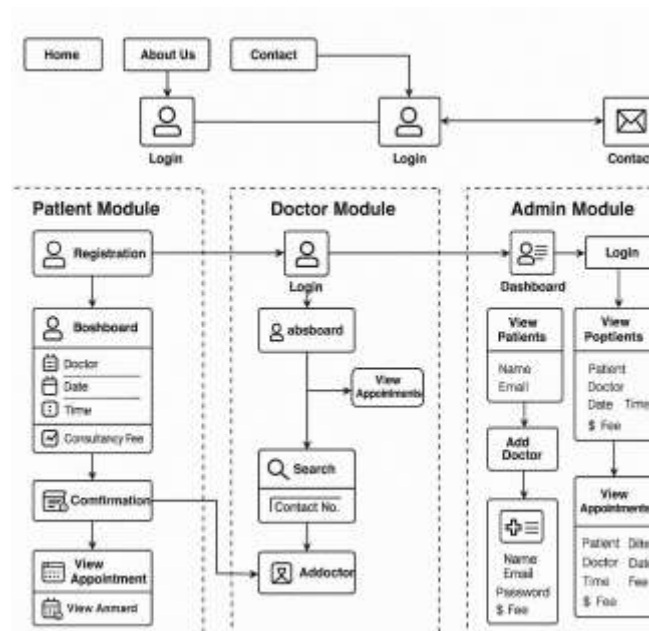


Figure 1: System Architecture

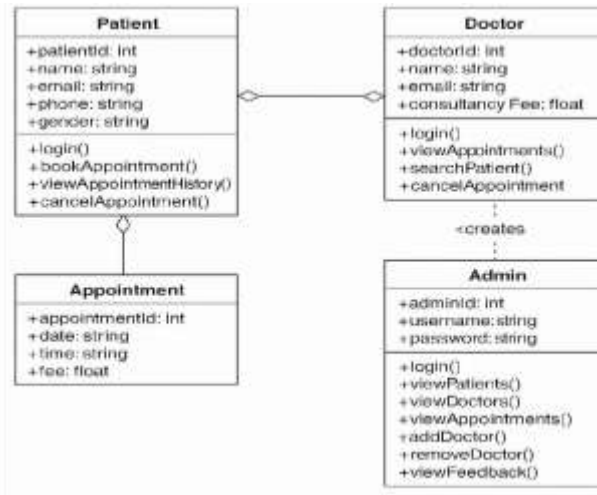


Figure 2: class Diagram

**3.2 Database Design:** Detail the database schema, including the tables used (e.g., patients, doctors, appointments, admin, contact queries) and their respective attributes. Explain the relationships between these tables using an Entity- Relationship Diagram (ERD).

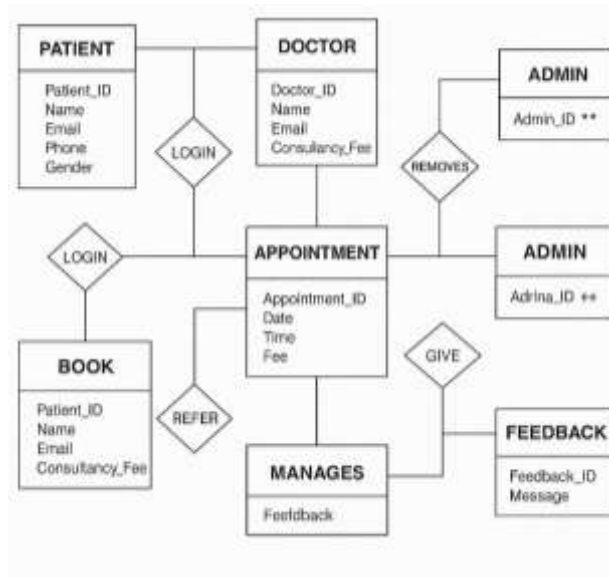


Figure 3: E-R diagram

**Home Page:** Has tabs or sections to let patients, doctors, and administrators log in. **About Us Page:** Provides information about the hospital, like its quality and services.

**Contact Page:** Enables users, including patients, to provide feedback or make inquiries to the hospital.

**Patient Module:**

**Registration:** Patients can create their own account by entering their name, email, phone number, password, and gender.

**Dashboard:** After logging in, patients see their personal control panel. **Book Appointment:**

Patients can choose a doctor, date, and time for their appointment.

They can also see how much the appointment will cost (consultancy fee). They get a confirmation message after booking.

**View Appointment History:** Patients can see a list of their past and upcoming appointments, including the doctor's name, fee, date, and time.

**Login:** If a patient already has an account, they can log in using their email and password to access their



dashboard. **Update:** Cancel Appointments: Patients can now delete their booked appointments. The system will show that the appointment was "deleted by you".

#### Doctor Module:

**Login:** Doctors can log in using a special form. **Dashboard:** After logging in, doctors see their schedule.

**View Appointments:** Doctors can see a list of appointments booked with them by patients, including the patient's name and the appointment time.

**Search Patient:** Doctors can search for a specific patient's appointment using the patient's contact number.

**Logout:** Doctors can log out of their account.

**Update:** Cancel Appointments: Doctors can also delete appointments from their schedule.

#### Admin Module:

**Login:** Administrators log in using a specific username and password ("admin", "admin123"). **Dashboard:** After logging in, administrators see an overview of the system.

**View Patients:** Administrators can see a list of all registered patients and their details (name, email, phone, password). They can also search for patients by phone number.

**View Doctors:** Administrators can see a list of all registered doctors and their details (name, email, password, consultancy fee). They can also search for doctors by email.

**View Appointments:** Administrators can see a complete list of all appointments, including patient details, doctor's name, date, time, and fee.

**Add Doctor:** Only administrators can add new doctors to the system by filling out a form with the doctor's name, email, password, and consultancy fee.

**View Feedback/Queries:** Administrators can read the messages and feedback sent by users through the "Contact" page.

**Logout:** Administrators can log out of their account.

**Update:** Remove Doctors: Administrators can now delete doctors from the system.

**3.3.1 Patient Module:** Describe the user interface (using HTML5, CSS3, Bootstrap) and the underlying PHP logic for registration, login, appointment booking, and viewing appointment history. JavaScript is key to making websites dynamic and ensuring valid form submissions. It achieves dynamic updates by directly changing the page's content and style after loading, often triggered by user events or server communication. For form validation, JavaScript checks user input against defined rules in the browser, offering immediate feedback.

**3.3.2 Doctor Module:** This area describes the steps doctors take to log in, the way their scheduled appointments are shown, and the search feature, which may utilize PHP and AJAX with JavaScript for enhanced interaction. Describes how the database is used to fetch and display appointment-related details.

**3.3.3 Admin Module:** Explain the login process and the functionalities for viewing patient and doctor lists, appointment details, adding doctors, and managing feedback. Describe the PHP scripts used for data retrieval, insertion, and deletion.

**3.4 User Interface Design:** Discuss the principles of user-centered design applied in the development, emphasizing the use of Bootstrap for a responsive and consistent layout.

#### Appendices:

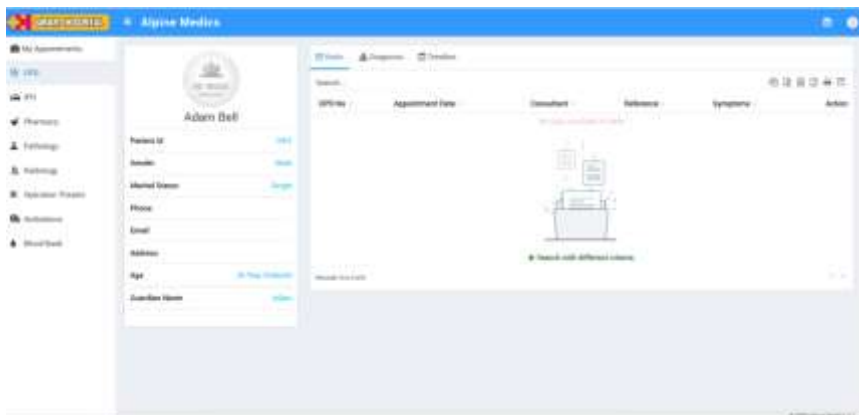
This section contains supplementary material that provides visual documentation of the Hospital Management System's user interface and key functionalities discussed throughout this paper. The figures presented here offer a visual reference for the system's design and layout as experienced by patients, doctors, and administrators.



A.1: Home Page



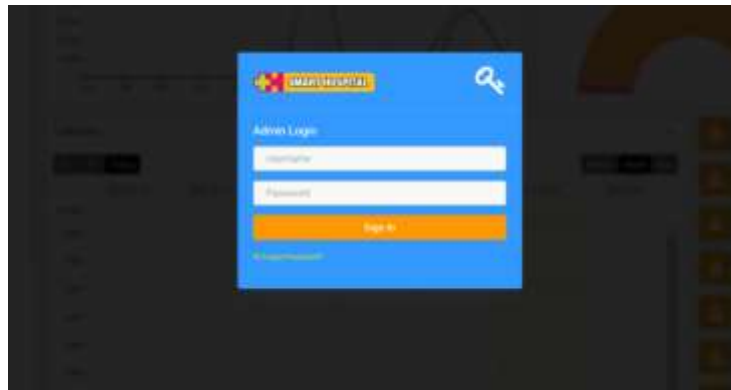
A.2: Patient Dashboard



A 3:OPD



A 4:In Patient Department



A 11:Admin login



A 12:Admin dashboard

#### IV. CONCLUSION

A web-based Hospital Management System built with PHP, MySQL, and modern front-end technologies offers an improved approach to healthcare administration. It provides patients, doctors, and administrators with a centralized and accessible platform for managing appointments, patient data, and system operations. The integration of features like online booking and administrative controls demonstrates the potential for increased efficiency and enhanced communication within a hospital setting. This system lays a foundation for future advancements in healthcare management through web-based solutions.

#### REFERENCES

- [1]. A. Sharma, P. Gupta, and R. Verma, "**Hospital Management System: A Digital Approach to Healthcare Administration,**" *International Journal of Computer Science & Healthcare Informatics*, vol. 5, no. 3, pp. 45-52, 2022.
- [2]. M. R. Patel and S. K. Desai, "**Cloud-Based Hospital Management System: A Secure and Scalable Approach,**" in *Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI)*, New York, USA, 2021, pp. 120-126.
- [3]. L. Wang, J. Kim, and T. Chen, "**AI-Powered HMS for Smart Hospitals,**" *Springer Journal of Artificial Intelligence in Healthcare*, vol. 12, no. 2, pp. 89-105, 2023.
- [4]. H. Singh and D. Roy, "**Blockchain in Healthcare Management: A Case Study on HMS Security,**" *Elsevier Journal of Medical Informatics*, vol. 28, no. 4, pp. 300-315, 2022.
- [5]. Bhattacharjee, S., & Ray, P. K. (2014). "Hospital Information Systems: A Review of Literature." *Health Policy and Technology*, 3(2), 77-84. Garg, A. X., et al. (2005). "Effects of Computerized Clinical Decision Support Systems on Practitioner Performance and Patient Outcomes: A Systematic